Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 38 (2012) 253 – 263

Procedia Engineering

www.elsevier.com/locate/procedia

International Conference on Modeling Optimisation and Computing (ICMOC 2012)

Genetic Algorithm with Search Bank Strategies for University Course Timetabling Problem

A. Araisa Mahiba[a], C. Anand Deva Durai[b]

[a] *Department of Computer Science and Engineering, Karunya University, Coimbatore- 641114, India*
[b] *School of Computer Science and Technology, Karunya University, Coimbatore- 641114, India*

Abstract

University Course Timetabling Problem (UCTP) is a multi-assignment problem in which Students, Staff, and Subjects together as events are scheduled to suitable timeslots and assigned to available classrooms. The design of scheduling of Course Timetable of a University is a difficult task every semester. This paper proposes a new method of Genetic algorithm with Search Bank Strategies namely local, guided and tabu searches. Local Search is used to increase the offspring or solutions. Guided Search is used to narrow the solutions by using Events Data Structure. Tabu search is used to remove the used solutions. The newly proposed method gives promising results for UCTP.

Keywords: Genetic Algorithm; Local Search; Guided Search; Tabu Search;

1. Introduction

Timetabling is one of the common scheduling problems which we see in day to day life. It is a multi-dimensional assignment problem in which set of Subjects, Students, Staff, Classrooms and timeslots are scheduled according to given constraints depends on the specific problem. Also Timetabling problem is a NP- Hard problem i.e. No smart algorithm is possible that leads to a simple or rapid solution and no known deterministic polynomial time algorithm for UCTP. An Efficient solution of timetable problem of specific domain will not work well for some other domain problem because it is depending upon the constraints specified.

Many algorithms are proposed to solve the timetable problem. In earlier days, graph coloring method is used. It works well for small instances but as the number of instances increased, it could not work efficiently. Researchers proposed a number of algorithms and approaches to solve UCTP by using graph coloring methods, constraint-based methods, population-based approaches (e.g., genetic algorithms

E-mail address: araisamahiba@karunya.edu.in.

1877-7058 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2012.06.033

(GAs), ant-colony optimization, and memetic algorithms), metaheuristic methods (e.g., tabu search (TS), simulated annealing (SA), and great deluge), variable neighborhood search (VNS), hybrid and hyper heuristic approaches, etc.

2. Related Works

There are two types of meta heuristics algorithms used namely local-area-based algorithms and population-based algorithms. Each type has its own advantages and disadvantages. Local-area-based algorithms concentrate on exploitation i.e. they search in one direction without finding all the possible solutions. E.g. Stimulated Annealing, very large neighbourhood search, Tabu Search, etc. This kind of algorithm improves the quality of solution. Population-based algorithm starts with number of solutions and refines the solutions to get optimum solution. So this is called as global-area-based algorithms. It requires more time and premature convergence. E.g. evolutionary algorithm, particle swarm optimization, ant-colony optimization, artificial immune system, etc. Various combinations of local-area-based and global-area-based algorithms are reported to solve timetabling problem.

In recent years, genetic algorithms are proposed to solve university-timetabling problem. Genetic algorithms are refined to improve the performance by modified genetic operators, heuristics operators and local search techniques.

A comprehensive review and recent research directions in timetabling can be found in [2], [3], and [4]. GAs has been used to solve the UCTP in the literature [5], [6], [7]. Rossi-Doria *et al.* [8] compared different metaheuristics to solve the UCTP. They concluded that conventional GAs do not give good results among a number of approaches developed for the UCTP. Hence, conventional GAs need to be enhanced to solve the UCTP. To solve this problem efficiently, we need to combine both the exploration ability and exploitation ability[9].

3. University Course Timetabling Problem

3.1. Problem Description

University Course Timetabling Problem is a multi-dimensional assignment problem, in which Staff, Subjects, a class of Students, timeslots and class rooms are scheduled according to the norms and rules of the University. This Timetable needs to satisfy some constraints before scheduling. Some are called Hard Constraints because these constraints must not be violated at any cost. E.g., a staff cannot be allocated to two classes at the same timeslot. Some constraints are called soft constraints because these constraints should preferably be satisfied, but can be accepted with a penalty associated to their violation. E.g., a staff should not have more than 3 consecutive classes of timeslots. Here some of the hard and soft constraints according to norms and rules of our university. The following are the hard constraints:

1. A Staff will not be assigned to two or more class of students at the same timeslot.
2. Two or more Staff should not be assigned to a class of students at the same timeslot.
3. Two or more class of students should not be allocated to a room at the same timeslot.
4. Two or more rooms should not be allocated to a class of students at the same timeslot.

Here are some soft constraints particularly for University Course Timetabling problem.

1. Staff should not have three consecutive classes of timeslots other than lab sections.
2. For consecutive timeslots, class of students should not change their class rooms.
3. All the timeslots should be occupied.

E-mail address: araisamahiba@karunya.edu.in.

3.2. Chromosome Structure and Formation

The heart of the genetic algorithm is the Chromosome. The Chromosome represents a potential solution and is divided into multiple genes. Genes represent distinct aspects of the solution as a whole. During the evolution process, chromosomes are exposed to multiple genetic operators that represent mating, mutation, etc. and then are chosen for the next generation during a natural selection phase based upon their "fitness," which is a measure of how optimal that solution is relative to other potential solutions. The makeup of chromosomes, which includes how many genes want and what those genes will represent. As given in the Fig. 1, the factors that make up the solution in university course timetabling problem are follows

- A set of N staff who taught different subjects, S= $\{s_1, s_2, s_3, \ldots, s_n\}$.
- A set of l Subjects to be taught, SB= $\{sb_1, sb_2, \ldots, sb_l\}$.
- A set of k Class of Students C=$\{c_1,c_2,c_3 \ldots c_k\}$.
- A set of (6 hrs x 6 days) Timeslots T= $\{t_1,t_2,t_3,\ldots t_{36}\}$.
- A set of m available rooms R=$\{r_1,r_2,r_3 \ldots r_m\}$.

Staff	Subjects	Students	Timeslots	Rooms

Fig. 1 Chromosome Structure

$$CHR_x = \{s_a, sb_b, c_c, t_d, r_e\}.$$

Fig. 2 Individual Chromosome

Where,

CHR_x represent xth chromosome
s_a represent ath staff which is integer which represent particular staff
sb_b represent bth subject which is integer which represent particular subject
c_c represent cth Class which is integer which represent particular Class
t_d represent dth Timeslot which is integer which represent particular Timeslot
r_e represent eth Room which is integer which represent particular Room

Each gene will be represented by integer values, which corresponds to particular staff, subject, class of students, timeslot and room. As shown in the Fig. 2, each gene will have values between lower and higher bound value in the chromosome which is potential solution. E.g. $CHR_{10}=\{0,4,6,7,3\}$.

Table 1 Factors and their present higher bound values

Factors	Representation	Taken into account
Timeslot per Day(TD)	X	6
Days per Week(DW)	Y	6
Subjects per Semester(SS)	Z	5
Staff(S)	N	15
Subjects(SB)	K * Z	6 * 5 = 30
Class of Students(C)	K	6
Timeslots(T)	X * Y	6 * 6 = 36
Rooms(R)	M	6

E-mail address: araisamahiba@karunya.edu.in.

3.3. Data Structure

Before creating the timetable and evaluating the fitness of the solution, we need the details of the staff who are handling which subjects and what class of students they are teaching. In addition, we need to know the subject is theory or lab, and how many credits are given to the subject because this is needed to allocate it a number of times per week. E.g., if the credit of a subject is 4, then the subject will be allocated 5 times per week (credit+1 times per week).

Normally this subject allocation will be done manually according to the priority (experience) of staff. Here we are having a 2-dimensional array, which consists of Staff, Subjects, Class of Students, and number of credits and flag, which denotes whether the subject is lab or theory. This is together called Events. This can be obtained from the user.

0 0 0 4 0
1 1 0 4 0
2 2 0 2 1
3 3 0 2 1
4 4 0 3 0
5 5 1 3 0
6 6 1 2 0

Fig. 3 Sample set of Events(Staff, Subjects, Class of Students, Credit of Subject and flag to note lab or theory)

In addition to the events array, there are few other arrays like staff, subject and room to store the staff, subject and classroom, which is placed in the timetable slots. Moreover some arrays are used to store temporary values and counts.

3.4. Fitness Function

A typical genetic algorithm requires:

1. A genetic representation of the solution domain,
2. A fitness function to evaluate the solution domain.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent.

The fitness function is a single method that must be implemented that accepts a potential problem solution and returns an integer value that indicates how good (or "fit") that solution is relative to other possible solutions. The higher the number, the better the solution. The lower the number (0 being the lowest legal fitness value), the poorer the solution. This will be used to find these fitness measurements to evolve the population of solutions toward a more optimal set of solutions.

The fitness function is calculated in two steps are following,

- Calculating the weightage of hard constraints satisfaction (HCS).
- Calculating the weightage of soft constraints satisfaction (SCS).

$$F(s) = \Sigma\, HCS + \Sigma\, SCS$$

First all potential solutions are evaluated for hard constraints satisfaction and the solutions, which satisfy, are called feasible solutions. Those feasible solutions will be evaluated for soft constraint satisfaction and which solutions satisfy is called most fitted solutions, which will be selected for next generations.

E-mail address: araisamahiba@karunya.edu.in.

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions (usually randomly) and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators.

4. A proposed method for University Course Timetabling Problem

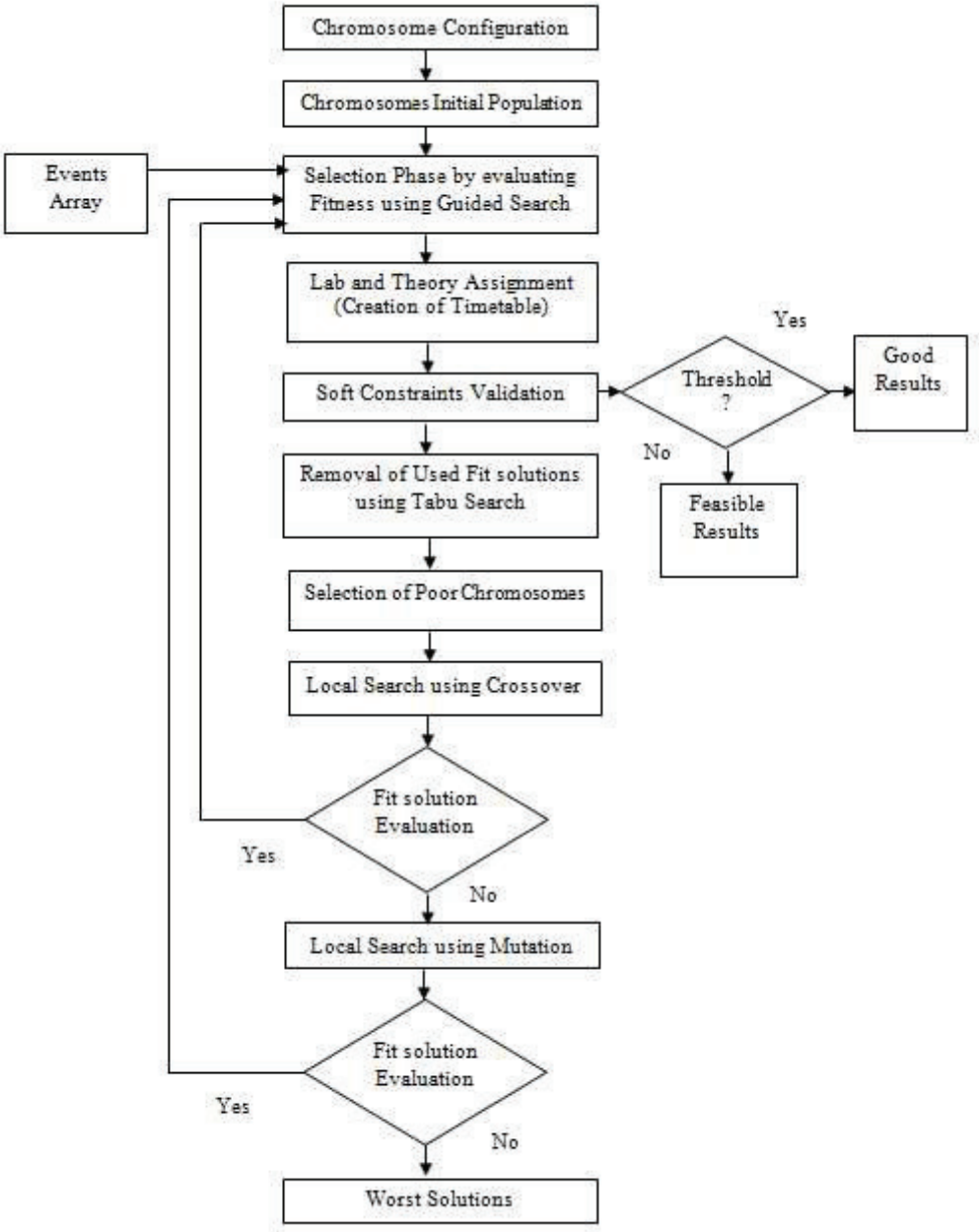


Fig. 4 Flow Chart of Proposed Method

E-mail address: araisamahiba@karunya.edu.in.

4.1. Chromosome Configuration and Initial Population

Chromosome is divided into 5 individual genes namely Staff, Subjects, Students, Timeslot and Classrooms. Each gene value is an integer, which can be mapped to Staff, Subjects, etc. First gene in the chromosome represents Staff, second represent Subject and fifth gene represents classroom. Each gene will have different limits i.e. lower and upper bound. E.g. Staff gene has lower bound 0 and upper bound 30. Limits need to be configured before generating the population .

Once all the above configuration is done, initial population is created. Initial population will be created by random. i.e. Each gene in each chromosome will be assigned a random value in between the limits. Population should be greater than the value of multiplication of upper bound of all the genes. E.g. upper bound of staff, subjects, class of students, timeslot and classrooms (15*30*6*36*6=583200), then population size should be more than 583200 i.e. 600000. Initially Fitness value of each chromosome should be assigned to zero.

4.2. Evaluating Fitness Function

In order to evaluate the fitness function, we are making use of the events array which contains Staff priority over subjects. We are comparing the alleles of genes (Staff, Subject and Class of Students) of Chromosomes in the population with the events in the array. Chromosomes which are same as the events are given fitness value 10. Most of the hard constraints are validated and fit solutions are obtained. Now fit solutions are obtained by checking the fitness value which is 10. Fig. 5 shows the chromosomes matched from the event array which contains first column Staff, second column subject, third column Class of students, fourth column timeslot, fifth column room and sixth column fitness value. Fig. 6 shows each events has matched how many chromosomes from the population.

	0	0	0	34	4	10.0
	0	0	0	29	4	10.0
	0	0	0	19	4	10.0
	0	0	0	25	3	10.0
	0	0	0	5	3	10.0
no:5						
	1	1	0	21	5	10.0
	1	1	0	16	3	10.0
	1	1	0	13	0	10.0
	1	1	0	5	1	10.0
	1	1	0	8	4	10.0
	1	1	0	28	1	10.0
no:23						

Fig. 5 Sample set of Chromosomes which are same as Events, their count and fitness value

Events:	0	1	2	3	4	5	6	7	8	9
No of Times:	4	19	30	26	22	49	48	53	38	51

Fig. 6 Sample set of Events, their count

E-mail address: araisamahiba@karunya.edu.in.

4.3. Creating Timetable

Creation of timetable consists of two parts namely lab assignment and theory subject assignment. First labs will be allocated in the timetable and then the theory subjects will be allocated. Theory and Labs in the event array will be differentiated by the flag present in it.

Labs are allocated by the chromosomes whose fitness value is 10 and that chromosome will be assigned to the continuous 3 timeslots.

To create Overall Timetable we are making use of array of 6x36 (Class of Students X Timeslot). Altogether 6*36 = 216 slots. A chromosome, which is having fitness value 10, fills each slot and count of each subject within a week is lesser than credit + one. Chromosomes that are used to fill the timeslot of the timetable are given additional fitness value 10. i.e. totally 20.

Subject Count is maintained by separate array for each subject. Since there are 30 subjects and each subject is given credit+1 slots per week, the slots filled will be 150(maximum). So each class will have 6 free timeslots per week.

Free timeslots are represented as */*/*. Filled timeslot is represented as 1/2/3. i.e. (Staff/Subject/Room). Room is allocated as each class has one room. Fig. 7 shows overall timetable for day order 1 and Fig. 8 shows the sample set of subjects and their occurrence in the timetable.

	0	1	2	3	4	5
Class:0	*/*/*	4/4/0	*/*/*	0/0/0	*/*/*	*/*/*
Class:1	*/*/*	*/*/*	7/7/1	8/8/1	7/7/1	6/6/1
Class:2	*/*/*	11/11/2	10/10/2	*/*/*	*/*/*	13/13/2
Class:3	*/*/*	1/16/3	0/15/3	1/16/3	4/19/3	*/*/*
Class:4	7/22/4	6/21/4	*/*/*	*/*/*	*/*/*	*/*/*
Class:5	10/25/5	11/26/5	*/*/*	11/26/5	12/27/5	*/*/*

Fig. 7 Overall Timetable of Day order 1

Subjects:	0	1	2	3	4	5	6	7	8	9
No of Times:	5	5	5	5	5	5	5	5	5	5

Fig. 8 Sample set of Subjects and their count.

4.4. Guided Search

Guided Search is one, which guides the search for creating the timetable. Guided search is implemented by using events array and subjects count array. For creating the Timetable, we are making use of events array and subjects count array and depending upon the array, fitness value is given to the chromosomes. The chromosomes, which are having highest fitness value, are used in the timetable slot.

4.5. Soft Constraint Validation and Results

The output of previous stage is creation of Timetable. Chromosomes which makes up the timetable will be having fitness value 20. The timetable will be validated for soft constraints. If timetable satisfies soft constraints, chromosomes which makes up will be awarded fitness value as 10 i.e. total 30. Along with overall timetable, separate timetable for each class of students and staff will be generated. Every generation, these outputs will be displayed to the user. Every generation, steps from evaluating the fitness value of chromosomes to mutation and worst solutions will be repeated according to the number of generations. But in first generation initial random population of chromosomes are used.

E-mail address: araisamahiba@karunya.edu.in.

4.6. Removal of Used Fit Solution

Chromosomes which are used in making timetable or having maximum fitness value will be selected and assign their fitness value as -1 so that these chromosomes will not be used again to create timetable. This is called tabu search, which removes already used solution or chromosome to avoid local minima.

4.7. Selection of Poor Chromosomes

Selection of poor chromosomes means selection of chromosomes having fitness value as 0 but not -1 because chromosomes with fitness value -1 will not be considered at all. Chromosomes having fitness value 0 will undergo crossover and/or mutation to increase as many as new chromosomes. This is also called local search.

4.8. Local Search using Crossover

Crossover is used to increase the number of new chromosomes. This is called local search because it increases new chromosomes using neighborhood moves. Here in Crossover, two types of neighborhood moves are used. First is among the poor chromosomes, two chromosomes will be randomly selected as undergo crossover by randomly selecting gene to swap that gene between two chromosomes. Thus, two new chromosomes will be generated.

Next neighborhood is by selecting three randomly selected chromosomes. Here also a gene will be randomly selected and that gene value will be swapped between three chromosomes thus producing three new chromosomes.

After these new chromosomes are created, it will be evaluated fitness by using events data structure. If it matches with events data structure, the chromosomes match will be awarded fitness value as 10.

If chromosomes are fit, then it will be selected for next generation to produce good timetable results. Otherwise it will undergo mutation which is another means to generate new chromosomes.

4.9. Local Search using Mutation

Mutation is also called Local search because it produces new solutions. Here a random poor chromosome i.e. fitness value is zero, is selected and a random gene is selected from that chromosome. A random value will be generated within the limits of that gene. Thus a new Chromosome will be produced. Then the new chromosome will be evaluated using events structure and fitness value will be increased accordingly. If the fitness value is improved, then it will be selected for next generation, Otherwise remaining chromosomes will not be used and those are called worst solutions.

5. Experimental Study

In this section, we experimentally analyze the performance of proposed method Search Bank Genetic Algorithm (SBGA) along with Tabu Search (TS), Steady State Genetic Algorithm (SSGA), Genetic Algorithm with Local Search (GALS), Guided Search Genetic Algorithm (GSGA), and Extended Guided Search Genetic Algorithm (EGSGA).

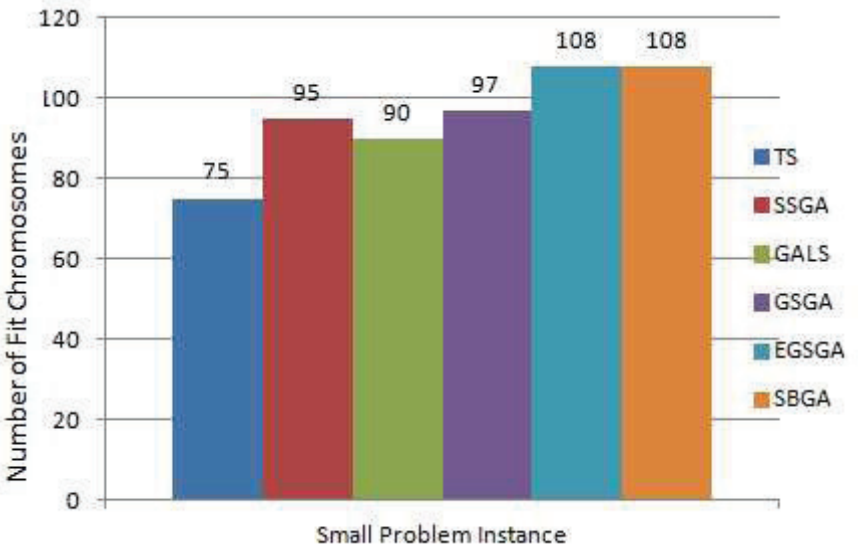
To test the performance, three groups of problem instances are chosen namely small, medium and large. For each groups, factors and range are chosen which is given in table 2. From each group, one instance of inputs is selected and given to the various algorithms. Each instance has 50 runs.

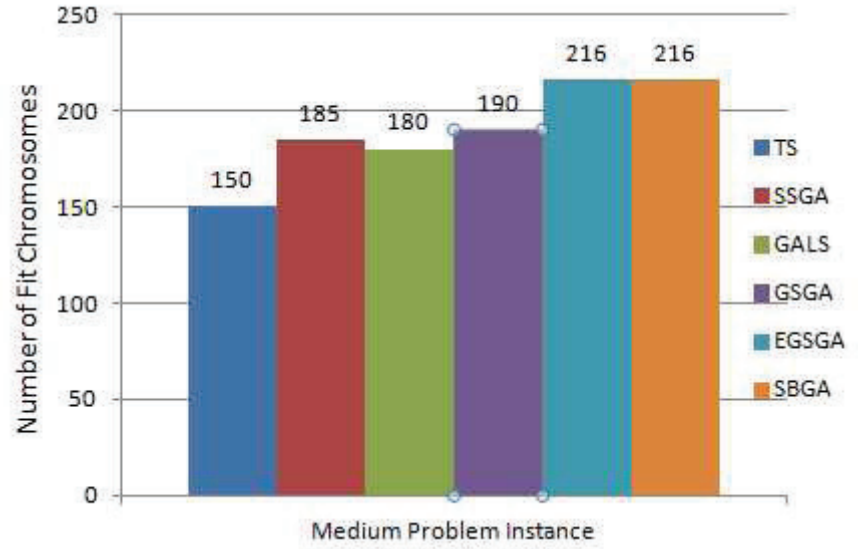
The metric to measure the performance is number of fit solutions used in each generation and it's average value is stored from all the methods.

E-mail address: araisamahiba@karunya.edu.in.

Table 2 Factors and their range for testing their performance

Factors	Small Instances	Medium Instances	Large Instances
Timeslot per Day(TD)	6	6	6
Days per Week(DW)	6	6	6
Subjects per Semester(SS)	5	5	5
Staff(S)	7	15	22
Subjects(SB)	3 * 5 = 15	6 * 5 = 30	9 * 5 = 45
Class of Students(C)	3	6	9
Timeslots(T)	6 * 6 = 36	6 * 6 = 36	6 * 6 = 36
Rooms(R)	3	6	9





E-mail address: araisamahiba@karunya.edu.in.

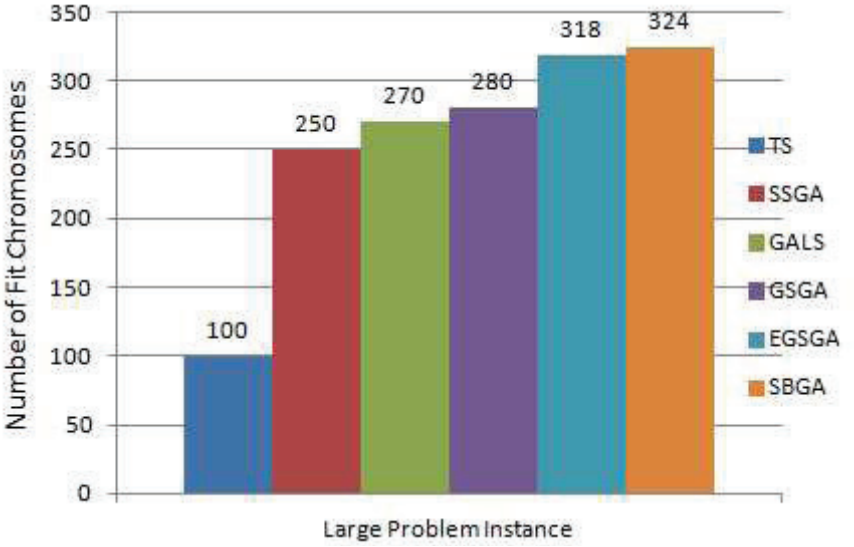


Fig. 9. Comparison of SBGA (Search Bank Genetic Algorithm) with other algorithms regarding the average performance on (a) small instances, (b) medium instances, and (c) large instances

The above three graphs plotted in the fig. 9 shows the average fit used solutions during generations in all the methods. When we look at the graphs, in all the instances (small, medium and large), maximum usage of slots (small-108, medium-216, and large-324) in the timetable occurred in the newly proposed Search Bank Genetic Algorithm. It means the fit used solutions will have fitness value forty including soft constraints satisfaction.

Another metrics to evaluate the performance is Time. For each run of an algorithm on a problem, the maximum run time t_{max} was measured as 10-12 s for small instances, 40-45 s for medium instances, and 60-70 s for the large instance based on the fact that larger UCTP instances are more complex and have more conflicting constraints and a larger search space as compared to smaller UCTP instances, and therefore, requires more processing time.

Although these problem instances lack many of the real-world problem constraints and issues, they allow the comparison of our approaches with current state-of-the-art techniques on them.

6. Conclusion

The timetabling problem consists of scheduling a sequence of lectures between teachers and students in a prefixed period of time (typically a week), satisfying a set of constraints of various types.

Chromosome structure, fitness function, hard constraints, soft constraints and data structure to evaluate violation of constraints are found for our University Course Timetabling Problem.

This new proposed method make use of local, guided and tabu search to make it more efficient. Local Search is used to produce as many as new possible solutions using mutation and crossover. Guided Search is used to direct the search using Events data Structure for better solutions. Tabu search are used to remove used solutions to avoid local minima. All these are being implemented for large-scale inputs and got promising results.

The future work can be inventing new genetic operators and adding hard and soft constraints as much as possible according to the norms of university.

E-mail address: araisamahiba@karunya.edu.in.

References

[1] S. N. Jat, S.Yang. Genetic Algorithms with Guided and Local Search Strategies for University Course Timetabling. *Proc. 4th Multidisciplinary Int. Conf. Scheduling Theory Appl; 2011,* p. 93 - 106.
[2] M. W. Carter, G. Laporte. Recent developments in practical course timetabling. in *Proc. 2nd Int. Conf. Pract. Theory Automated Timetabling*, (Lecture Notes in Computer Science) 1998; vol. 1408, p. 3–19.
[3] R. Lewis. A survey of metaheuristic based techniques for university timetabling problems. *OR Spectrum* 2008; vol. 30, no. 1, p. 167–190.
[4] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot. A survey of search methodologies and automated system development for examination timetabling. *J. Sched.* 2009; vol. 12, no. 1, p. 55–89.
[5] S. Abdullah, H. Turabieh. Generating university course timetable using genetic algorithm and local search. in *Proc. 3rd Int. Conf. Hybrid Inform. Tech.* 2008; p. 254–260.
[6] R. Lewis, B. Paechter. Application of the grouping genetic algorithm to university course timetabling. in *Proc. 5th Eur. Conf. Evol. Comput. Combinat. Optim.* (Lecture Notes in Computer Science) 2005; vol. 3448, p. 144–153.
[7] P. Pongcharoen, W. Promtet, P. Yenradee, C. Hicks. Schotastic optimization timetabling tool for university course scheduling. *Int. J. Prod. Econ.*, 2008; vol. 112, no. 2, p. 903–918.
[8] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, T. St¨utzle. A comparison of the performance of different metaheuristics on the timetabling problem. in *Proc. 4th Int. Conf. Pract. Theory Automated Timetabling* (Lecture Notes in Computer Science), 2003; vol. 2740, p. 329–351.
[9] S. N. Jat, S.Yang. A guided search genetic algorithm for the university course timetabling problem. in *Proc. 4th Multidisciplinary Int. Conf. Scheduling: Theory Appl.*, 2009; p. 180–191.
[10] J.-F. Cordeau, B. Jaumard, R. Morales. Efficient timetabling solution with tabu search. *the 2002 International Timetabling Competition (TTC2002)*, 2003.

E-mail address: araisamahiba@karunya.edu.in.